Algorithmic complexity of pair cleaning method for k-satisfiability problem. (draft version)

Sergey Kardash

April 18, 2012

Abstract

It's known that 3-satisfiability problem is NP-complete. Here polynomial algorithm for solving k-satisfiability ($k \geq 2$) problem is assumed. In case theoretical points are right, sets P ans NP are equal.

1 Introduction

Definition 1. *Formulae $A(x)$ is called k-CNF if*

$$A(x) = \bigcap_{i=1}^{n} \bigcup_{j=1}^{k} x_{u_{ij}}^{\sigma_{ij}}, \sigma_{ij} \in \{0,1\}, u_{ij} \in \{1,\cdots,m\}, \forall i \in \{1,\cdots,n\}, \forall j \in \{1,\cdots,k\}$$

$\bigcap$ - *conjuntion operation,*
$\bigcup$ - *disjuntion operation,*
m - *number of variables in formulae,*
n - *number of clauses,*
k - *number of variables in each disjunction,*
n_t - *number of clause groups.*

$$x^\sigma = \begin{cases} x, \sigma = 0 \\ \bar{x}, \sigma = 1 \end{cases}$$

Example 1. *3-CNF $A(x) = (x_1 \cup x_2 \cup x_3) \cap (\bar{x_1} \cup x_3 \cup \bar{x_4})$. Here $m = 4$, $n = 2$, $k = 3$, $n_t = 2$.*

Definition 2. *Let formulae $A(x)$ is k-CNF. Problem of defining whether equation $A(x) = 1$ has solution or not is called k-satisfiability problem of formulae A(or k-SAT(A)).*

Example 2. *k-satisfiability problem of formulae A described in Example 1 (k-SAT(A)) is defining whether $\exists x \in B^m$ (boolean vector of size m): $A(x) = 1$. It's evident that $x_0 = (1,1,1,1)$ makes $A(x_0) = 1$. $A(x_0)$ is satisfiable. k-CNF $B(x) = (x_1 \cup x_2) \cap (\bar{x_1} \cup x_2) \cap (x_1 \cup \bar{x_2}) \cap (\bar{x_1} \cup \bar{x_2})$ is an example of not satisfiable task. There is no $x_0 : A(x_0) = 1$. On the contrary $A(x) = 0, \forall x$.*

It was proved that 2-satisfiability problem has polynomial solution (by Krom [2]). We are going to show polynomial algorithm(from n) for any $k-SAT$. By the way we describe method of getting 1 explicit solution of corresponding equation $A(x) = 1$ in case source task is satisfiable which is polynomial from n and method of solving equation $A(x) = 1$ which is polynomial from number of such solutions.

2 Method description

Initially new mathematic objects and operations for them are introduced. After description of method in pure mathematic way algorithmic presentation which is more readable is given. Almost each structure has 2 common structures associated with it: 1)variable set associated with this structure and 2)some value sets of these variables. Though they will be defined separately it's easy to see common logic of their introduction.

Let $x_{s_1 s_2 \cdots s_k} = (x_{s_1}, x_{s_2}, \cdots, x_{s_k})$. Further in order to avoid enumeration of variables which are not related to described structure we list important variables using such notation.

Definition 3. *Clause group signed $T_{s_1 s_2 \cdots s_k}(A)$ is a set of all clauses $\bigcup_{j=1}^{k} x_{s_j}^{\sigma_{t_j}}$ where $u_{i1}u_{i2}\cdots u_{ik} = s_1 s_2 \cdots s_k$. Variable set associated with $T_{u_{s_1} u_{s_2} \cdots s_k}(A)$(or $X(T_{s_1 s_2 \cdots s_k}(A))$) is $x_{s_1 s_2 \cdots s_k}$. Value of clause group $T_{s_1 s_2 \cdots s_k}(A)$ is a value of $x_{s_1 s_2 \cdots s_k}$ such that k-CNF consisted of all clauses from clause group $T_{s_1 s_2 \cdots s_k}$ is equal to 1. Value set induced by clause group $T_{s_1 s_2 \cdots s_k}(A)$ (or $V(T_{s_1 s_2 \cdots s_k}(A))$) is a set of all values of this clause group.*

Example 3. *Though clauses $x_1 \cup x_2 \cup x_3$ and $\bar{x_1} \cup x_2 \cup \bar{x_3}$ have different degrees they belong to the same clause group T_{123} in case they present in formulae A.*

Example 4. *For example clause group T_{123} consists of clauses $x_1 \cup x_2 \cup x_3$ and $\bar{x_1} \cup x_2 \cup \bar{x_3}$. Value set induced by this clause group can be presented using table below:*

x_1	x_2	x_3
0	0	1
0	1	0
0	1	1
1	0	0
1	1	0
1	1	1

Each row corresponds to one value of x_{123} . We have excluded from this list only sets which make 3-CNF $(x_1 \cup x_2 \cup x_3) \cap (\bar{x_1} \cup x_2 \cup \bar{x_3})$ equal to 0 ($x_{123} = (0, 0, 0)$ and $x_{123} = (1, 0, 1)$).

Definition 4. *k-CNF A(x) all clauses of that can be classified into n_t clause groups is called k-CNF of degree n_t. It also can be signed as $A_k^n(x)$or $A_k(x)$ or $A^n(x)$.*

Example 5. *2-SAT $A(x) = (x_1 \cup x_2) \cap (\bar{x_1} \cup x_2) \cap (x_2 \cup x_3) \cap (\bar{x_2} \cup \bar{x_3})$ has 2 clause groups T_{12} and T_{23}, so it's degree is 2 and it can be signed as $A_2^2(x)$ or $A_2(x)$ or $A^2(x)$.*

Definition 5. *Clause combination F for formulae $A(x)$ consisted from clause groups $T_{u_{i_1 1} u_{i_1 2} \cdots u_{i_1 k}}(A), T_{u_{i_2 1} u_{i_2 2}, \cdots, u_{i_2 k}}(A), \cdots, T_{u_{i_l 1} u_{i_l 2} \cdots u_{i_l k}}(A)$ (or $F(T_{u_{i_1 1} u_{i_1 2} \cdots u_{i_1 k}}, T_{u_{i_2 1} u_{i_2 2}, \cdots, u_{i_2 k}}, \cdots, T_{u_{i_l 1} u_{i_l 2} \cdots u_{i_l k}}, A)$) is a set of listed clause groups. Variable set associated with it is $x_{h_1 h_2 \cdots h_r}$ where each variable index from set of clause groups is presented only once.*

We'll deal with different value sets of variables associated with clause combination and in order not to confuse them let's write them out separately.

Definition 6. *Value of clause combination $F(T_{u_{i_1 1} u_{i_1 2} \cdots u_{i_1 k}}, T_{u_{i_2 1} u_{i_2 2}, \cdots, u_{i_2 k}}, \cdots, T_{u_{i_l 1} u_{i_l 2} \cdots u_{i_l k}}, A)$ is a value of $x_{h_1 h_2 \cdots h_r}$ - variable set associated with it such that k-CNF consisted of all clauses associated with listed clause groups equal to 1.*

Definition 7. *Value set of clause combination $F(T_{u_{i_1 1} u_{i_1 2} \cdots u_{i_1 k}}, T_{u_{i_2 1} u_{i_2 2}, \cdots, u_{i_2 k}}, \cdots, T_{u_{i_l 1} u_{i_l 2} \cdots u_{i_l k}}, A)$ based on $A(x)$ is a set of values of this clause combination.*

Definition 8. *Value set of clause combination $F(T_{u_{i_1 1} u_{i_1 2} \cdots u_{i_1 k}}, T_{u_{i_2 1} u_{i_2 2}, \cdots, u_{i_2 k}}, \cdots, T_{u_{i_l 1} u_{i_l 2} \cdots u_{i_l k}}, A)$ induced by $A(x)$ is a set of all values of this clause combination.*

It's easy to see that value set induced by clause combination $F(T_{u_{i_1 1} u_{i_1 2} \cdots u_{i_1 k}}, T_{u_{i_2 1} u_{i_2 2}, \cdots, u_{i_2 k}}, \cdots, T_{u_{i_l 1} u_{i_l 2} \cdots u_{i_l k}}, A)$ is a value set based on this clause combination.

Example 6. *Let we have 2 clause groups: $T_{12}(A)$ which has clauses $x_1 \cup x_2$ and $\bar{x_1} \cup x_2$ in formulae A and $T_{23}(A)$ which has clauses $x_2 \cup x_3$ and $\bar{x_2} \cup \bar{x_3}$. Then value set induced by clause combination $F(T_{12}, T_{23})$ is a set of all possible values of x_{123} which make 2-SAT $(x_1 \cup x_2) \cap (\bar{x_1} \cup x_2) \cap (x_2 \cup x_3) \cap (\bar{x_2} \cup \bar{x_3})$ equal to 1.*

x_1	x_2	x_3
0	1	0
0	1	1

Each row of the list is a value of clause combination $F(T_{12}, T_{23})$, i. e. $x_{123} = (0, 1, 0)$.

Definition 9. *Relationship structure for k-CNF $A(x)$ ($R(A)$) is a set of all possible clause combinations consisted of $(k + 1)$ clause groups.*

Example 7. *For 2-CNF $A(x) = (x_1 \cup x_2) \cap (x_1 \cup \bar{x_2}) \cap (x_2 \cup x_3) \cap (x_1 \cup \bar{x_3}) \cap (x_1 \cup x_4) \cap (\bar{x_1} \cup x_4)$ clause groups are: $T_{12}, T_{23}, T_{13}, T_{14}$. $R(A) = \{F(T_{12}, T_{23}, T_{13}), F(T_{12}, T_{23}, T_{14}), F(T_{12}, T_{13}, T_{14}), F(T_{23}, T_{13}, T_{14})\}$.*

Definition 10. *Value set of relationship structure induced by k-CNF $A(x)$ ($V_i(R(A))$) is a set of value sets of clause combinations induced by $A(x)$ involved in relationship structure based on k-CNF $A(x)$.*

Example 8. *For Example 7 value set of relationship structure induced by k-CNF $A(x)$ is a set of tables listed below:*

$V(F(T_{12}, T_{23}, T_{13}, A))$:

x_1	x_2	x_3
1	0	1
1	1	0
1	1	1

, $V(F(T_{12}, T_{23}, T_{14}, A))$:

x_1	x_2	x_3	x_4
1	0	1	1
1	1	0	1
1	1	1	1

, $V(F(T_{12}, T_{13}, T_{14}, A))$:

x_1	x_2	x_3	x_4
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

, $V(F(T_{23}, T_{13}, T_{14}, A))$:

x_1	x_2	x_3	x_4
0	1	0	0
0	1	0	1
1	0	1	1
1	1	0	1
1	1	0	1

$V_i(R(A)) = \{V(F(T_{12}, T_{23}, T_{13}, A)), V(F(T_{12}, T_{23}, T_{14}, A)), V(F(T_{12}, T_{13}, T_{14}, A)), V(F(T_{23}, T_{13}, T_{14}, A))\}$.

Definition 11. *Value set of relationship structure based on k-CNF $A(x)$($V_b(R(A))$) is a set of value sets of clause combinations based on $A(x)$ involved in relationship structure based on k-CNF $A(x)$*

Example 9. *For Example 7 value set of relationship structure based on k-CNF $A(x)$ is any set $V_b(R(A)) = (V_1, V_2, V_3, V_4)$ where $V_1 \subseteq V(F(T_{12}, T_{23}, T_{13}))$, $V_2 \subseteq V(F(T_{12}, T_{23}, T_{14}))$, $V_3 \subseteq V(F(T_{12}, T_{13}, T_{14}))$, $V_4 \subseteq V(F(T_{23}, T_{13}, T_{14}))$. In example:*

V_1:

x_1	x_2	x_3
1	1	0
1	1	1

, V_2:

x_1	x_2	x_3	x_4
1	1	1	1

, V_3:

x_1	x_2	x_3	x_4
1	0	0	1
1	0	1	1
1	1	0	1

, V_4:

x_1	x_2	x_3	x_4
0	1	0	0
0	1	0	1
1	0	1	1
1	1	0	1
1	1	0	1

Definition 12. *Value set of relationship structure based on k-CNF $A(x)$ is called empty ($V(R(A)) = \varnothing$) if at least one value set of clause combination value set of relationship structure consists of is empty.*

Definition 13. *Let R(A) - relationship structure for k-CNF $A(x)$. $V(R(A)) = \{V_1, V_2, \cdots, V_t,\}$, $G(R(A)) == \{G_1, G_2, \cdots, G_t,\}$ - 2 value sets of this relationship structures based on $A(x)$. We call $V(R(A))$ included in $G(R(A))$ (or $V(R(A)) \subseteq G(R(A))$) if $V_i \subseteq G_i, \forall i \in \{1, \cdots, t\}$.*

Example 10. *Let V(R(A)) is a set described in Example 9 and G(R(A)) is a set from example 8. $V \subseteq G$. Indeed all value sets of relationship structure based on k-CNF $A(x)$ are included in the value set of relationship structure induced by k-CNF $A(x)$.*

Definition 14. *Let we have 2 clause combinations $F(T_{i_1}, T_{i_2}, \cdots, T_{i_s}, A)$ and $F(T_{j_1}, T_{j_2}, \cdots, T_{j_r}, A)$. Let they have common variables $x_{i_1}, x_{i_2}, \cdots, x_{i_s}$ - those variables which present in both clause combinations. Clearing of given pair of value sets V_1 and V_2 of clause combinations $F(T_{i_1}, T_{i_2}, \cdots, T_{i_s}, A)$ and $F(T_{j_1}, T_{j_2}, \cdots, T_{j_r}, A)$ correspondingly based on k-CNF A(x) is a process of deleting $x^1_{a_1 a_1 \cdots a_z} \in V_1$ for which $\nexists x^2_{b_1 b_2 \cdots b_u} \in V_2 : x^1_{i_1 i_2 \cdots i_s} = x^2_{i_1 i_2 \cdots i_s}$ and deleting $x^2_{b_1 b_2 \cdots b_u} \in V_2$ for which $\nexists x^1_{a_1 a_1 \cdots a_z} \in V_1 : x^1_{i_1 i_2 \cdots i_s} = x^2_{i_1 i_2 \cdots i_s}$. Clearing procedure is briefly marked as $C(V_1, V_2)$.*

Example 11. *Let's take 2 values of clause combinations from Example 8:*

$V(F(T_{12}, T_{23}, T_{13}, A))$:

x_1	x_2	x_3
1	0	1
1	1	0
1	1	1

and $V(F(T_{23}, T_{13}, T_{14}, A))$:

x_1	x_2	x_3	x_4
0	1	0	0
0	1	0	1
1	0	1	1
1	1	0	1
1	1	0	1

.

Common variables are $x_{123} = (x_1, x_2, x_3)$. Let's explore table which corresponds to $V(F(T_{12}, T_{23}, T_{13}))$. $x^1_{123}(1) = (1, 0, 1)$ has corresponding $x^2_{1234}(3) = (1, 0, 1, 1)$(in brackets $x^2_{1234}(3)$, 3 is a number of row in the table) and it should be saved. $x^1_{123}(2)$ has even 2 corresponding rows: $x^2_{1234}(4)$ and $x^2_{1234}(5)$. But for last one, $x^1_{123}(3)$, we can't find corresponding values from second table with the same common variables and it should be deleted from values based on $V(F(T_{12}, T_{23}, T_{13}))$. The same should be done with $x^2_{1234}(1)$ and $x^2_{1234}(2)$. After clearing

$$V_1: \begin{array}{ccc} x_1 & x_2 & x_3 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \quad \text{and } V_2: \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{array}.$$

It can be briefly marked as $C(V(F(T_{12}, T_{23}, T_{13}, A)), V(F(T_{23}, T_{13}, T_{14}, A))) = (V_1, V_2)$.

Definition 15. *Clearing of value set of relationship structure (V_r) based on k-CNF $A(x)$ (pair cleaning method for formulae $A(x)$) is a process of clearing of all possible pairs of value sets of clause combination based on k-CNF $A(x)$ contained in V_r until clearing is impossible. We'll note result of cleaning as C(V(R(A))).*

Pair cleaning method in algorithmic form

$V_{new} \leftarrow V_{source}(R(A))$
repeat
  $V_{old} \leftarrow V_{new}$
  for $i = 1 \rightarrow d-1$ **do**
    for $j = i+1 \rightarrow d$ **do**
      $(V^i_{new}, V^j_{new}) \leftarrow C(V^i_{new}, V^i_{new})$
    end for
  end for
until $V_{new} = V_{old}$

where
d - number of clause combinations in relationship structure,
$V_{source}(R(A))$ - value set of relationship structure induced by $A(x)$.

Definition 16. *Let $V = V(R(A))$ - value set of relationship structure based on k-CNF $A(x)$. V is called unclearable if $V = C(V)$.*

Lemma 1. *Let $V = V(R(A))$ - value set of relationship structure induced by k-CNF $A(x)$. $V_{res} = C(V)$. $V_{res} \neq \varnothing \Leftrightarrow \exists\ V_1 \subseteq V_{res}$ where V_1 - unclearable value set of relationship structure based on k-CNF $A(x)$ where each value set of clause combination consists of 1 value of this clause combination.*

Proof. $\Rightarrow$
This can easily be proved using induction. We'll take induction not for clauses but for clause groups. In this proof n_t - number of clause groups. It's evident that $n_t \leq n$. In case $n_t \leq k+1$ statement is evident because cleaning of values of relationship structure is reduced to clearing the only clause combination.

Let the case $n^0_t = k+1$ be the basis of induction. Let's assume statement is right for $n_t > k+1$. We need to prove (n_t+1) case. Let $A^{n_t+1}(x)$ - source k-CNF (see Definition 4). $R = R(A)$ - relationship structure for it. V - value set of relationship structure induced by k-CNF $A(x)$.

Let $V_C = C(V)$ - result of pair clearing method which is not empty ($V_C \neq \varnothing$). After clearing relationship structure induced by k-CNF with (n_t+1) clause groups we have not empty value set of relationship structure. Let's choose any clause group T_{n_t+1} (we'll use both types of notation - $T_{i_1 i_2 \ldots i_k}$ which shows variables involved in clause group building and $T_j, j \in \{1, \cdots, n_t+1\}$) - a serial number of clause group from formulae $A^{n_t+1}(x)$. Let's look at $B^{n_t}(x)$ - formulae which has the same clause groups as $A^{n_t+1}(x)$ excluding T_{n_t+1}. Let R_B - relationship structure based on $B^{n_t}(x)$, V_B - value set of this relationship structure. It's evident that all clause combinations of R_B are clause combinations of R. Beside them R has clause combinations which contain T_{n_t+1} with all possible combination without repetition of k clause groups which are common for $A^{n_t+1}(x)$ and $B^{n_t}(x)$ (i. e. $F(T_{n_t+1}, T_1, T_2, \cdots, T_k)$).

Let's V_B has value sets of clause combinations the same as value sets of corresponding clause combinations of V_C. It's evident that $C(V_B) = V_B$. $V_C \neq \varnothing \Rightarrow V_B \neq \varnothing \Rightarrow$ exists $V^1_B \subseteq V_B$ where V^1_B - unclearable value set of relationship structure based on k-CNF $B^{n_t}(x)$ where each value set of clause combination consists of 1 value. (according to induction step). Now we need show that $\exists\ V^1_A \subseteq V_A$ - unclearable value set of relationship structure based on k-CNF $A^{n_t+1}(x)$ where each value set of clause combination consists of 1 value. This proof is very trivial.

Indeed, let's look at T_{n_t+1}(another notation for this clause group is $T_{l_{(n_t+1)1} l_{(n_t+1)2} \cdots l_{(n_t+1)k}}$). In this clause group there are 2 types of variables: those that present at least in one clause group $T_j, j \in \{1, \cdots, n_t\}$(common variables) and those that absent in this set. Let's explore first group (present). We can say that exists such clause combination $F(T_{n_t+1}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A)$ from relationship structure R where all common variables from T_{n_t+1} can be found at least in one of other members of this clause combination: $T_{i_1}, T_{i_2}, \cdots, T_{i_k}$. This statement can easily be proved by building this clause combination. Number of common variables can't be greater than k. So we can find corresponding clause group for each common variable which also contains this variable. Number of such clause groups is less or equal k and if it's less we add arbitrary clause groups in order to get clause combination which contains $k+1$ clause groups. And now let's build another clause combination $F(T_{i_{k+1}}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A)$

which has k common clause groups with $F(T_{n_t+1}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A)$ and $T_{i_{k+1}}$ is a clause group from $B^{n_t}(x)$ (this clause group can be found because $n_t > k+1$).

By the way we need prove that each variable of clause combination in unclearable value set of relationship structure where each value set of clause combination consists of 1 value has the same value in all clause combinations of that value of relationship structure. This result will also be used in next lemma. That's easy to be shown.

Let x_i - arbitrary variable presented in relationship structure. Let $F_1 = F(T_{x_i x_j \cdots})$ and $F_2 = F(T_{x_i x_z \cdots})$ - 2 different clause combinations which are parts of relationship structure R. V^1 - unclearable values of relationship structure where each value set of clause combination consists of 1 value. Let value $V_1^{F_1}$ of clause combination from V_1 which corresponds F_1 and value $V_1^{F_2}$ of clause combination from V_1 which corresponds F_2 have different value of variable x_i. Then operation C($V_1^{F_1}$, $V_1^{F_2}$) will give empty sets to both values. But that's contradiction because values of relationship structure is unclearable.

The fact that V_C is not empty and $V_B^1 \subseteq V_B$ means that value of clause combination $F(T_{i_{k+1}}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A)$ from V_B^1 is also a value of the same clause combination from V_B and from V_C. The fact that it can't be deleted during clearing means that exists value $V_{T_n}^B$ of clause combination $F(T_{n_t+1}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A)$ from V_C which has the same values of common variables as value of $F(T_{i_{k+1}}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A)$ from V_B^1. The only thing we need to prove now is that all clause combinations from V_C which contain T_{n_t+1} have value which can be added to V_B^1 and $V_{T_n}^B$ to create new value of relationship structure V_C^1 which is unclearable.

Let's notice that these clause combinations don't give any new variables to clause combinations of R_B and $F(T_{n_t+1}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A$ This fact and the fact that in V_B^1 all values of the same variables in different clause combinations are the same can give us a hint that value of each clause combination which contains T_{n_t+1} consisted of the same variable values as they presented in V_B^1 and value of clause combination $F(T_{n_t+1}, T_{i_1}, T_{i_2}, \cdots, T_{i_k}, A)$ discussed in previous paragraph.

$\Leftarrow$

This side is evident: the fact that $\exists\ V_1 \subseteq V_{res}$ means that $V_{res} \neq \varnothing$.

Lemma is proved. ∎

Lemma 2. *Let V_1 - value set of relationship structure based on k-CNF $A(x)$ where each value set of clause combination consists of 1 value of this clause combination. V_1 is unclearable $\Leftrightarrow$ k-CNF $A(x)$ is equal to 1 on this value set.*

Proof. $\Rightarrow$

It was proved in Lemma 1 that corresponding variables have the same values in different clause combinations. Let's have a glance at k-CNF which variables values are the same as in the structure. It's evident that such k-CNF is equal to 1. Indeed for each clause exists clause combination that involves this clause. Clause combination is equal to 1 on this set $\Rightarrow$ clause itself is equal to 1. All clauses on this set are equal 1 $\Rightarrow$ k-CNF value on this set is equal 1.

$\Leftarrow$

This proof is trivial. We take variable values $x_{12\cdots m}$ that make k-CNF equal 1. It's evident that in value set of relationship structure V_1 based on $A(x)$ each value set of clause combination which is a member of V_1 and has the same variable values as $x_{12\cdots m}$ is unclearable.

Lemma is proved. ∎

Theorem 1. *Result of pair cleaning method applied to source k-CNF is not empty $\Leftrightarrow \exists$ solution of equation $k - CNF = 1$.*

Proof. Consecutive usage of Lemma 1 and Lemma 2 proves the theorem. ∎

Theorem 2. *Let*

V - value set of relationship structure based on k-CNF $A(x)$,

$V_C = C(V)$ - cleared value set of relationship structure,

V_C^1 - unclearable value set of relationship structure based on k-CNF $A(x)$ where each value set of clause combination based on k-CNF consists of 1 value,

V_{F_i} - value set of clause combination F_i,

$V_{F_i}^C$ - values of clause combination F_i^C,

$V_{F_i}^0$ - value of clause combination F_i.

Then $V_{F_i}^0 \in V_{F_i}$ - member of $V_C = C(V) \Leftrightarrow \exists V_C^1 : V_{F_i}^0 \in V_{F_i}^C$ - member of V_C^1

Proof. Scheme of proof is the same as for Lemma 1, it's full description will be given a bit later. ∎

So we have not only algorithm for solving k-satisfiability problem but also algorithm for solving equation $A(x) = 1$. Of course in common case it's not polynomial (because number of solutions is $O(2^n)$). But process of getting each root of equation is polynomial. We'll describe it in full preprint version of this paper.

3 Complexity

Number of values clause group can take is less than 2^k.
Number of values clause combination can take is less than $2^{k(k+1)}$.
Number of clause combinations in relationship structure is $C_{n_t}^{k+1}$.
Number of comparisons during one iteration pass is less than $2^{2k(k+1)}(C_{n_t}^{k+1})^2$.
Number of iterations is less than $2^{k(k+1)}C_{n_t}^{k+1}$.
That means that number of operations for algorithm is less than $2^{3k(k+1)}(C_{n_t}^{k+1})^3$.
Therefore complexity of $k-SAT$ is $O(n_t^{3(k+1)})$. For 3-SAT it's $O(n_t^{12})$.
$2^{-k}n \leq n_t \leq n \Rightarrow$ method's complexity is $O(n^{3(k+1)})$. For 3-SAT it's $O(n^{12})$. That means that pair cleaning method is polynomial and P=NP.

References

[1] Cook, Stephen (April 2000). The P versus NP Problem . Clay Mathematics Institute. Retrieved 2006-10-18.

[2] Krom, Melven R. (1967), "The Decision Problem for a Class of First-Order Formulas in Which all Disjunctions are Binary", *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik*, **13**, pp. 15-20.